REMARKS

The above-referenced patent application has been reviewed in light of the Office Action of December 17, 2003. Reconsideration of the above-referenced patent application in view of the amendments and remarks is respectfully requested.

No claims have been amended or added. Applicants believe that this response addresses the Examiner's rejection and that any changes do not introduce new matter into the specification, limit the scope of the claims or result in any prosecution history estoppel.

RESPONSE TO 35 U.S.C. §103 REJECTION

The Examiner rejected claim 18 under 35 U.S.C. §102(e) as being anticipated by Rivard et al. (U.S. Patent No. 6,300,953) in view of Duluk (U.S. Patent No. 5,596,686). Applicants respectfully disagree with the Examiner's assertion. In particular, the Examiner refers only to column 6, lines 47-67 and figure 10 of Rivard to support his assertion:

> "Referring next to FIG. 10, each tag module 610 comprises four tag entry modules tag0 to tag3 and one tag control module 615. Each tag entry includes an address tag and an index number from 0 to 3. Upon reset, tag0 resets its index to 0, tag1 resets its index to 1, and so on. Each tag entry also resets to a "not valid" state. Mapping from an address tag to an entry in cache 300 is therefore made through the illustrated index signal. A Least Recently Used ("LRU") mapping policy is preferably implemented by shifting the tag and index towards the bottom of the list until the tag is replaced with new data at the top of the list. When an entry hits, the tag is collapsed in the list and re-inserted at the top of the list. Tag controller 615 coordinates the four tag entry modules. It should be noted that a Least Recently Allocated ("LRA") policy may be implemented in the alternative, but LRU is preferred."

Rivard discloses the usage of tags but none of the specifics claimed. In particular, nowhere does the above passage or figure 10 teach or suggest the specifics of: (a) receiving texture addresses for a first pixel, checking if the addresses match the addresses in a first stage of a multi-stage cache controller and doing one of the following, (1) loading the addresses in the first stage if there is no valid address in the first stage (2) reloading the addresses in the first stage if a match is found or (3) moving to a second stage if no match is found; (b) if step (a)(1) is true transferring the corresponding texture data from main memory into cache memory with a first tag; (c) if step (a)(2) is true, making no transfer of texture data because data has already been transferred; (d) if step (a)(3) is true, checking if the addresses match the addresses in the

second stage and doing one of the following (1) if there is no addresses in the second stage moving the addresses from the first stage to the second stage and loading the addresses into the first stage (2) if a match is found moving the addresses from the first stage to the second stage and loading the addresses into the first stage (3) moving to a third stage if no match is found; (e) if step (d)(1) is true transferring corresponding texture data from the main memory into the cache memory with a second tag; (f) if step (d)(2) is true making no transfer of texture data because data has already been transferred; (g) if step (d)(3) is true, repeating step (d) for subsequent stages and using subsequent tags where necessary, until a last stage been checked or until a match has been found; (h) if the last stage has been checked and no match found loading the addresses into the first stage and moving the stored addresses to the next stage in sequence and overwriting the addresses from the last stage; and (i) if step (h) is true transferring corresponding texture data from the main memory into cache memory with the tag of the last stage addresses; (j) wherein when addresses are loaded into the first stage the tag assigned will be either the tag of the last stage or the tag within the stage that was hit, and transferring texture data at a main memory access granularity and so forth.

Duluk also fails the teach or suggest the above, alone or in combination with Rivard.

RESPONSE TO 35 U.S.C. §103 REJECTION

The Examiner rejected claims 2-9, 13, 14, 16-17 under 35 USC 103(a) as being unpatentable over Gannett (U.S. Patent No. 5,790,130).

CLAIM 19 AND DEPENDENT CLAIMS

Applicants respectfully disagree with the Examiner's assertion. In particular, Gannett fails to teach or suggest "said cache controller including a plurality of least recently used controllers coupled in succession to thereby transfer texels according to a least recently used replacement algorithm" as claimed in claim 19 and the claims which depend therefrom. As provided for in the specification:

> The present invention includes multiple cache controllers that implement a one clock least recently used algorithm for the efficient scheduling of texels to be mapped onto the current pixel and issuing a fetch command for the required data from the main memory.
>
> The cache memory control system of the present invention also includes a cache read/write arbiter that efficiently transfers the data coming from main

> memory into the texture cache. It also uses a look ahead mechanism to schedule the data out of the cache memory to the interpolating filter relative to the data being loaded into the cache in order to create a system that can sustain the peak output of one value per clock.

No where does Gannett disclose or suggest a cache controller including a plurality of least recently used controllers coupled in succession to thereby transfer texels according to a least recently used replacement algorithm. The passages the Examiner in his office action do not teach or suggest the above. In particular, Gannett at column 9, lines 7-20 and column 21, line 64 to column 22, line 6 merely provides:

> In one embodiment of the present invention, the local memory of the hardware device is arranged as a cache memory in which portions of textures are stored in the local memory of the hardware device at any one time. The device independent portion of TIM tracks the usage of the portions of texture data stored in the cache and monitors the priorities of the textures to predict future usage of those portions. A cache miss in the hardware occurs when texture data is needed by the hardware device to render an image that is not currently stored in the cache. When a cache miss occurs, TIM determines which block of texture data within the cache to replace by considering which block or blocks of texture data within the cache were used least recently and which textures have the lowest priority.
>
> * * *
>
> As described above, each pixel can potentially map to four texels from one MIP map, or eight texels from multiple MIP maps. As discussed in more detail below, texel data downloaded to the cache is organized in the main memory of the host computer by TIM so that any four adjacent texels in each MIP map are located in separate interleaves so that they can be accessed in parallel. Thus, any four adjacent texels in a MIP map that may be needed to generate resultant texel data through bilinear interpolation can be read in a single read operation. When trilinear interpolation is employed, the two sets of four texels from adjacent MIP maps can be read in two read operations.

CLAIM 14 AND DEPENDENT CLAIMS

Additionally, Gannett fails to teach or suggest "said texture cache arbiter transfers said texels from said texture main memory into the cache memory according to a look-ahead algorithm to hide read and write access clock cycles between sequential pixels" as claimed in claim 14 and the claims that depend therefrom. As noted in the application:

> Figure 8 is a block diagram of the cache arbiter 92. The cache arbiter 92 controls the loading through write control 102 and accessing through read control 100 of the cache memory 30 based on the pixel texturing requirements. These pixels reside in the pipeline. The read FIFO 94 provides a stream of pixels that are needed to be texture mapped. Stored in FIFO 94 is enough information to tell what texels are needed for each pixel and which controllers sections W, X, Y, and Z of the cache need to be loaded prior to the processing of the current pixel.
>
> The read fifo 94 has enough depth to hide the latency of the texture access from main memory. Enough stages of pipelining within the cache arbiter 92 are provided so that texture data per pixel can be loaded sequentially ahead of when the pixel actually reads texture data from the cache. As the data for a given pixel enters the arbiter 92, all fetch or read data requests to move data into the cache are completed unless there is a conflict with cache locations for pixels with pending read request that are still in the pipeline. In the event of a conflict, the loading of the data is stalled until the pixels in front of the current load are sent to the interpolator process and the conflict over the cache location has been removed. Also, if the data necessary for the pixel to be sent to texture interpolation process has not been loaded into the cache when the pixel has reached the last pipe stage, it will be held there until all the data needed has been loaded into the cache.

No where does Gannett disclose or suggest a texture cache arbiter that transfers texels from the texture main memory into the cache memory according to a look-ahead algorithm to hide read and write access clock cycles between sequential pixels. The passages the Examiner in his office action do not teach or suggest the above. In particular, Gannett at column 21, line 64 to column 22, line 6 merely provides:

> As described above, each pixel can potentially map to four texels from one MIP map, or eight texels from multiple MIP maps. As discussed in more detail below, texel data downloaded to the cache is organized in the main memory of the host computer by TIM so that any four adjacent texels in each MIP map are located in separate interleaves so that they can be accessed in parallel. Thus, any four adjacent texels in a MIP map that may be needed to generate resultant texel data through bilinear interpolation can be read in a single read operation. When trilinear interpolation is employed, the two sets of four texels from adjacent MIP maps can be read in two read operations.

In view of the above, the claims are asserted to be allowable over Gannett. Applicants respectfully request the claims be allowed.

CONCLUSION

In view of the foregoing, it is respectfully asserted that all of the claims pending in this patent application are in condition for allowance.

The required fee for a three month extension of time is enclosed. No additional fees are required for additional claims. Should it be determined that an additional fee is due under 37 CFR §§1.16 or 1.17, or any excess fee has been received, please charge that fee or credit the amount of overcharge to deposit account #02-2666.

If the Examiner has any questions, he is invited to contact the undersigned at (323) 654-8218. Reconsideration of this patent application and early allowance of all the claims is respectfully requested.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR, & ZAFMAN LLP

Dated: July 26, 2005

By _____
Farzad E. Amini, Reg. No. 42,261

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, Post Office Box 1450, Alexandria, Virginia 22313-1450 on July 26, 2005.

_____
Margaux Rodriguez *July 26, 2005*